Instalando e Configurando o Squid

Este guia não pretende ser definitivo, ele somente é utilizado para fins de ensino. Qualquer configuração adicional é bom olhar no próprio manual que acompanha o software.

Criado por: Marcos Aurelio Pchek Laureano
marcos@laureano.eti.br

Última alteração: 07/11/2002

Considerações

- Os testes para configuração e utilização foram feitos utilizando a distribuição Suse Linux (versão 8.0) e no Red Hat Linux (versão 8.0);
- Sinta-se à vontade para copiar qualquer coisa daqui, mas por favor, cite a origem.

Conceitos de Servidores Proxy

O objetivo principal de um servidor proxy é possibilitar que máquinas de uma rede privada possam acessar uma rede pública, como a Internet, sem que para isto tenham uma ligação direta com esta. O servidor proxy costuma ser instalado em uma máquina que tenha acesso direto à Internet, sendo que as demais efetuam as solicitações através desta. Justamente por isto este tipo de servidor é chamado de proxy, pois é um *procurador*, ou seja, sistema que faz solicitações em nome dos outros.

Um servidor proxy para o protocolo http, por exemplo, pode ter outras funcionalidades implementadas. Visto que todas as solicitações de páginas efetuadas pelas máquinas da rede privada serão feitas através dele, é muito útil armazenar localmente as páginas que foram solicitadas, permitindo que os próximos acessos, efetuados por quaisquer máquinas da rede, possam ser otimizados. Este conceito é chamado de *caching*, e vários servidores proxy na verdade são servidores proxy cache. Pelo mesmo motivo, também é possível implementar uma funcionalidade que permita controlar o que os clientes podem acessar e em que momento. Um servidor proxy também pode implementar o NAT (*Network Address Translation* – Tradução de Endereços de Rede). O NAT é tecnicamente a função de um portal de nível de rede, mas alguns fornecedores também incluem este recurso em seus produtos e servidores proxy.

Proxy Caching

Os servidores de proxy cache são implementados na camada de aplicativo e processam protocolos Internet específicos, tais como http e FTP. São definidas regras no servidor proxy para determinar como um pedido de estação de trabalho deve ser processado.

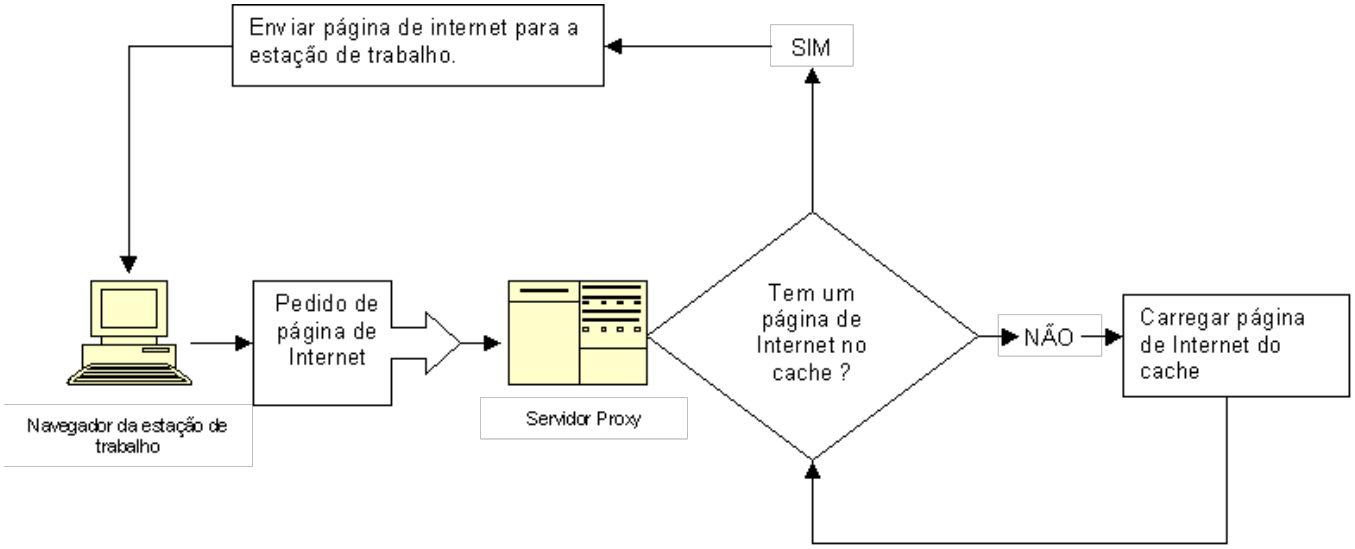
Uma das principais tarefas de um servidor proxy é armazenar temporariamente páginas da Web e arquivos de FTP para clientes proxy. Esses tipos de servidores proxy são chamados de servidores de cache proxy. O cache aumenta o desempenho da rede ao reduzir a quantidade de dados que são transferidos de fora da rede local.

Para implementar o proxy caching, cada estação de trabalho da rede é configurada como um cliente proxy para um determinado serviço. Por exemplo, um cliente proxy Web iria configurar seu navegador (*browser*) para reconhecer o servidor proxy. Quando um cliente fizer um pedido no navegador para baixar uma certa página, o navegador fará o pedido ao servidor proxy. O servidor proxy contém armazenadas as páginas visitadas recentemente. Este cache contém as páginas Web que as estações de trabalho em toda a rede baixaram recentemente.

O servidor proxy verifica o seu cache para ver se a página da está disponível. Se a página estiver disponível no cache será enviada ao cliente a página armazenada. Se a página não estiver no cache, o servidor proxy baixará do site em questão, armazenará essa página no seu cache e a enviará à estação de trabalho.

Para garantir que as páginas no cache não estejam desatualizadas, os dados do cache proxy expiram após um tempo pré-determinado. No squid, esta configuração é chamada de *tempo de renovação* de objeto (som, vídeo, arquivos texto, etc...).

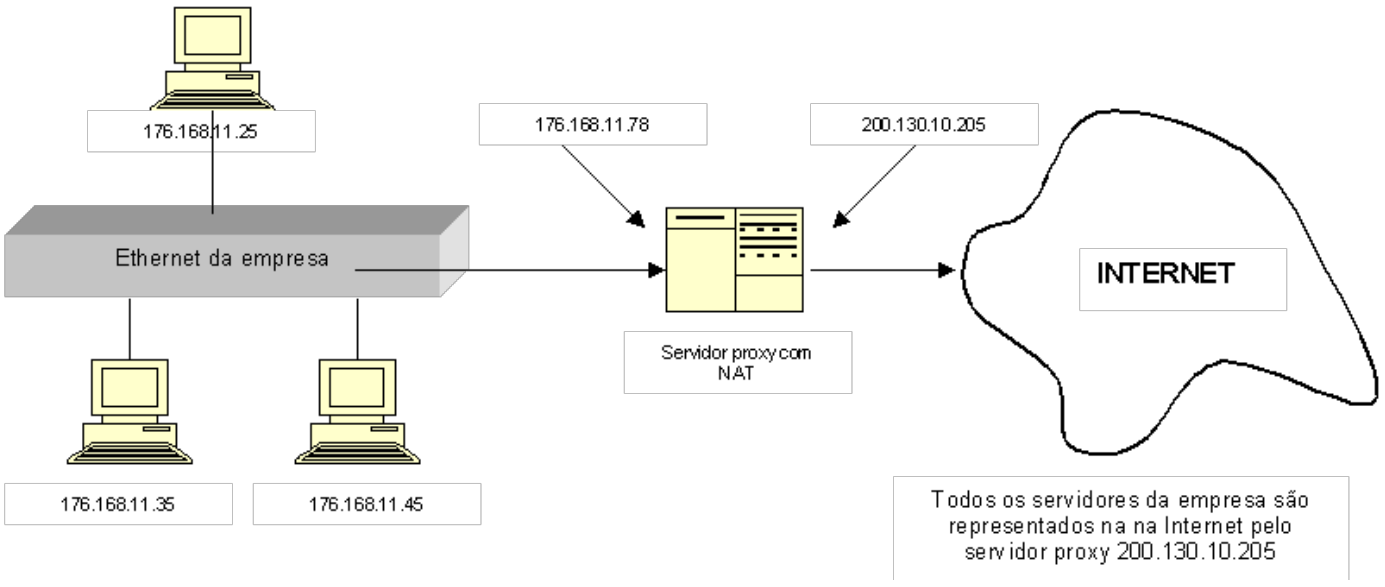
Este processo aumenta o desempenho da rede porque a página é baixada imediatamente para o cliente a partir do servidor proxy, evitando ter de baixá-la da Internet.



Tradução de Endereços de Rede.

Muitos servidores proxy são distribuídos com a função de NAT. A NAT permite que o endereço de rede interno de uma empresa seja ocultado da Internet. A empresa é representada na Internet como um endereço de IP não relacionado com os endereços de IP internos.

Utilizando um servidor proxy, todo o tráfego de dentro da empresa destinado à Internet é enviado para o servidor proxy. O proxy dá cada pacote um outro endereço de IP antes transmiti-lo pela Internet. Quando o pacote de resposta chega, o servidor proxy o envia ao servidor apropriado da empresa que havia feito o pedido. Este procedimento protege os endereços verdadeiros da rede interna da Internet, dificultando o ataque de um hacker ao sistema, já que o endereço do sistema protegido não é conhecido e não fica diretamente acessível a partir da Internet.



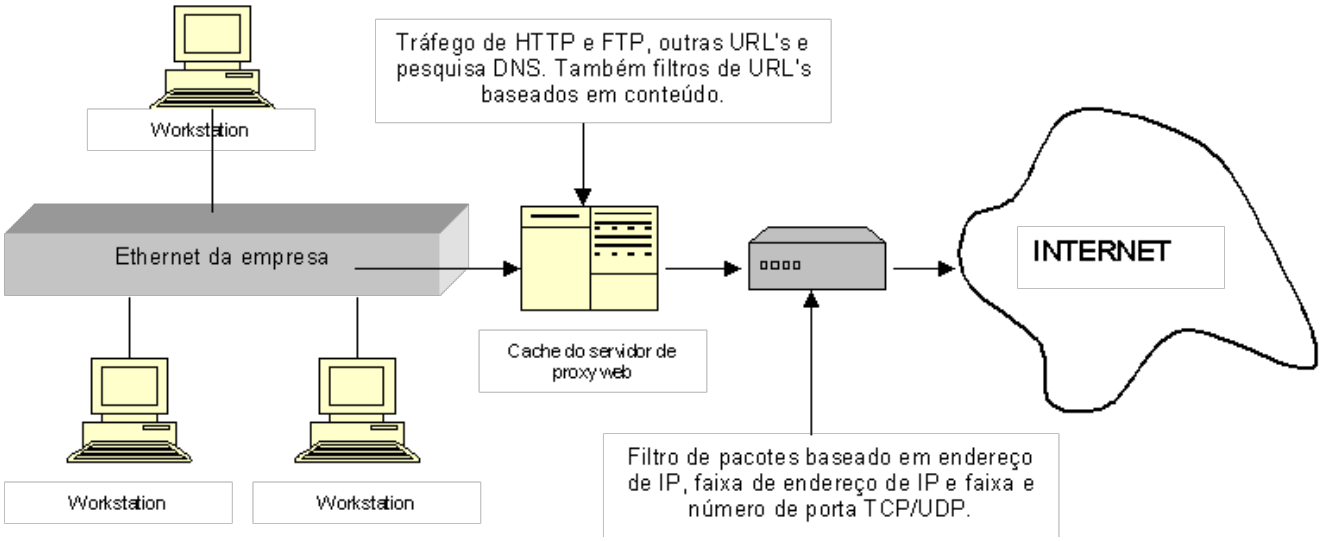
Diferença Entre um Filtro de Pacotes e um Servidor Proxy

Os 2 serviços são colocados no perímetro da rede. Ambos funcionam como um intermediário entre uma LAN e a Internet. Ambos possuem a capacidade de filtrar o tráfego transmitido para dentro e para fora da rede. Ambos utilizam regras para determinar se certos tipos de tráfego terão autorização para passar pelo servidor ou se serão descartados.

O que acontece é que o filtro de pacotes não penetra tão fundo no pacote como o servidor proxy. O filtro de pacotes analisa o tráfego nas camadas de rede (camada 3) e de transporte (camada 4) do modelo OSI. Por exemplo, um filtro de pacotes determinará se autoriza a passgem de um endereço ou de uma certa faixa de endereços IP. Se a sua rede é atacada a partir de uma faixa constante de endereços de IP, você pode criar uma regra para descartar todos os pacotes que se originarem dessa faixa. Você pode filtrar o tráfego por

serviço ou número de porta (por exemplo), criando uma regra que descarte todo o tráfego direcionado a certas portas de escuta, tais como FTP, rlogin e tráfego Telnet, ou dentro de uma faixa de números de portas. Você pode filtrar pacotes ICMP por tipo ou código, o que permite descartar somente certos tipos de tráfego ICMP. Isto ajuda na sua proteção contra ataques comuns de *denial-of-service* distribuídos (DDOS). Como os filtros de pacotes trabalham nessas camadas, são muitas vezes implementados em roteadores no perímetro da rede.

O servidor *proxy* é capaz de analisar pacotes na camada de aplicativo (camada 7). Isto oferece uma flexibilidade muito maior, porque permite que o tráfego dentro de um serviço, como o tráfego da porta 80 (http) possa ser filtrado. Como mencionado anteriormente, isto permite que os servidores proxy analisem o tráfego http ou FTP, e determinem se deve ou não passar. Se houver uma regra que impeça a passagem de qualquer endereço WEB que contiver a palavra “sexo”, então qualquer pedido de URL http que contiver “sexo” será descartado.



O SQUID

O servidor Squid Web Proxy Cache é gratuito e funciona em código aberto para Unix e Linux. Ele permite que os administradores implementem um serviço de *proxy caching* para Web, acrescentem controles de acesso (regras), e armazenem até mesmo consultas de DNS.

O Squid originou-se de um programa desenvolvido pelo projeto *Harvest* chamado *cached* (*Cache Daemon*). A *National Science Foundation* (NSF) financia o desenvolvimento do Squid através do *National Laboratory of Network Research* (NLANR).

O Squid é um *Web proxy cache* que atende à especificação HTTP 1.1. É utilizado somente por clientes proxy, tais como navegadores Web que acessem à Internet utilizando HTTP, Gopher e FTP. Além disso, ele não trabalha com a maioria dos protocolos Internet. Isto significa que ele não pode ser utilizado com protocolos que suportem aplicativos como vídeo-conferência, *newsgroups*, *RealAudio*, ou videogames como o *Quake* ou *Counter Strike*. O principal motivo destas limitações é que o Squid não é compatível com programas que utilizem UDP. O Squid usa o UDP somente para comunicação inter-cache.

Qualquer protocolo de cliente suportado pelo Squid deve ser enviado como um pedido de proxy no formato HTTP. A maioria dos navegadores suporta esta função, portanto, os protocolos FTP, HTTP, SSL (*Secure Socket Layer*), WAIS (*Wide Area Information Server*) são suportados na maioria das redes que utilizam o Squid.

Os protocolos funcionarão se você os solicitar utilizando o seu navegador e se ele estiver configurado como um cliente *proxy* para o servidor *Web proxy cache*.

O Squid também suporta protocolos internos e de administração. Tais protocolos são usados entre os caches que puderem existir em outros no mesmo ou em outros servidores de *proxy-caching*, ou para a administração de um proxy cache.

Internet Cache Protocol (ICP) – Consulta outros caches sobre um determinado objeto;

Cache Digest – Obtém um índice de objetos de outros caches;

HTTP – Obtém os objetos de outros caches;

Hypertext Caching Protocol (HTCP) – Está sendo incluído no Squid;

Simple Network Management Protocol (SNMP) – Obtém informações sobre o *proxy* e as envia a uma *Network Management Station* (NMS) para análise.

Requisitos de Sistema Específicos para o Proxy Caching

O Squid utiliza mais recursos de sistema do que outros aplicativos. Os dois principais subsistemas de hardware que o Squid utiliza e deve ter um bom desempenho é o tempo de busca aleatória e a quantidade de memória no sistema.

Tempo de busca aleatória em disco – Para um proxy cache, o tempo de busca aleatória deve ser o mais baixo possível. O problema é que os sistemas operacionais procuram aumentar a velocidade de acesso em disco utilizando vários métodos que geralmente reduzem o desempenho do sistema;

Quantidade de memória do sistema – A memória RAM é extremamente importante para a utilização de um proxy cache. O Squid mantém uma tabela na memória RAM sobre os seus objetos. Se uma parte dessa tabela tiver que sofrer swapping, o desempenho do Squid será bastante degradado. O Squid é um processo, então qualquer swapping tornará o programa mais lento. Por exemplo, se você tiver 16 GB armazenados no cache, precisará de 96 MB (aproximadamente) de RAM para o indíce de objetos.

Outros requisitos do sistema, como velocidade de CPU, não são tão importantes assim. A velocidade do processador somente será notado durante o início do sistema (durante a criação do indíce de objetos). Um sistema multiprocessado não constuma fazer diferença no desempenho do proxy cache, pois o Squid contém uma pequena porção de código encadeado.

Instalando o Squid

Você pode baixar a versão binária (código executável) que acompanha a sua distribuição, mas o ideal é baixar os códigos fontes do site do Squid e compilar.

Procure a última versão para *download* (quando este manual foi escrito a última versão disponível era 2.5.STABLE1) e baixe o arquivo *squid-2.5.STABLE1.tar.gz*.

É necessário criar um usuário para squid. É com este usuário que o squid irá ser ativado. Algumas distribuições já criam o usuário por padrão (depende do perfil de instalação solicitado)

Execute o comando finger para verificar se o usuário já está cadastrado:

```
# finger squid
```

Deverá surgir informações parecidas com esta:

```
Login: squid Name: (null)
Directory: /var/spool/squid Shell: /dev/null
Never logged in.
No mail.
No Plan.
```

Se o comando finger não funcionar, você pode utilizar o comando grep.

```
# grep squid /etc/passwd
```

Se aparecer informações como abaixo, o usuário já está cadastrado.

```
squid:x:23:23::/var/spool/squid:/dev/null
```

Execute os seguintes comandos para cadastrar um usuário:

Execute os seguintes comandos para cadastrar um usuário:

```
# groupadd squid
# useradd –g squid –s /dev/null squid >/dev/null 2>&1
```

Execute o comando (copie o arquivo para o diretório */tmp* ou algum outro da sua preferência):

```
# tar zxvf squid-2.5.STABLE1.tar.gz
```

Neste momento você deve ter um diretório chamado *squid.2.5.STABLE1* no seu diretório atual.
Agora temos que compilar e instalar o Squid, acesse o diretório e execute os comandos.

```
# ./configure --prefix=/usr/local/squid
# make all
# make install
```

Configuração do Squid

Utilizando o arquivo */etc/squid/squid.conf* vamos configurar o Squid. Este arquivo define as configurações, tais como o número da porta HTTP em que o Squid ouvirá os pedidos HTTP, pedidos de entrada e saída, informações de timeout e dados de acesso ao firewall. O arquivo foi criado durante a instalação do Squid.

O arquivo squid.conf é definido com as configurações padrão do Squid e pode ser utilizado após várias modificações. É necessário realizar as alterações, pois por padrão, o squid.conf nega o acesso a todos os navegadores. O Squid será completamente inútil até que você faça as alterações no arquivo.

Cada opção de configuração no squid.conf é identificada como uma *tag*. Cada *tag* é uma configuração do Squid. Por exemplo, a definição de porta de pedido de cliente HTTP é identificada pela *tag http_port*. O arquivo squid.conf estará localizado no diretório /usr/local/squid/etc (no meu exemplo), se você instalar através de um pacote binário o arquivo pode estar no diretório /etc ou /etc/squid. O arquivo de configuração do squid é auto-documentável, ou seja, todas as tags possuem uma explicação sobre a configuração. Veja o exemplo:

```
# TAG: http_port
# Usage: port
# hostname:port
# 1.2.3.4:port
#
# The socket addresses where Squid will listen for HTTP client
# requests. You may specify multiple socket addresses.
# There are three forms: port alone, hostname with port, and
# IP address with port. If you specify a hostname or IP
# address, then Squid binds the socket to that specific
# address. This replaces the old 'tcp_incoming_address'
# option. Most likely, you do not need to bind to a specific
# address, so you can use the port number alone.
#
# The default port number is 3128.
#
# If you are running Squid in accelerator mode, then you
# probably want to listen on port 80 also, or instead.
#
# The -a command line option will override the *first* port
# number listed here. That option will NOT override an IP
# address, however.
#
# You may specify multiple socket addresses on multiple lines.
#
# If you run Squid on a dual-homed machine with an internal
# and an external interface then we recommend you to specify the
# internal address:port in http_port. This way Squid will only be
# visible on the internal address.
#
#Default:
# http_port 3128
```

Por padrão, todas as linhas do Squid estão desabilitadas. Para habilitar, devemos retirar o caracter # que aparece antes da *tag*. Se você não modificar o arquivo de configuração, o Squid rodará com as configurações padrões.

TAG VISIBLE_HOSTNAME

Esta tag identifica o servidor do proxy, a configuração padrão é *none*, neste caso o Squid pegará o retorno da função *gethostname().* Caso apareça alguma mensagem de erro, você deverá configurar esta tag.

A mensagem de erro:
```
FATAL: Could not determine fully qualified hostname. Please set 'visible_hostname'
Squid Cache (Version 2.5.STABLE1): Terminated abnormally.
CPU Usage: 0.008 seconds = 0.006 user + 0.002 sys
Maximum Resident Size: 0 KB
Page faults with physical i/o: 238
Aborted
```

Onde está:
```
# get errors about IP-forwarding you must set them to have individual
# names with this setting.
#
#Default:
# none
```

Deixe:
```
# get errors about IP-forwarding you must set them to have individual
# names with this setting.
#
#Default:
# none
visible_hostname nomedoseuservidor
```

Se não aparecer nenhuma mensagem, o squid está executando. Para confirmar, execute o comando:
```
# ps aux | grep squid
```

TAG HTTP_PORT

Esta tag configura a porta HTTP onde o Squid ouve os clientes *proxy*. A porta padrão é a 3128. A porta 8080 também costuma ser utilizada. É possível configurar as duas portas para o Squid aceitar as requisições.

Abra o arquivo squid.conf procure a tag para realizar a configuração.

```
# internal address:port in http_port. This way Squid will only be
# visible on the internal address.
#
#Default:
# http_port 3128
```

Mude para:
```
# internal address:port in http_port. This way Squid will only be
# visible on the internal address.
#
#Default:
http_port 3128 8080
```

TAG CACHE_DIR

Esta *tag* define onde os dados do cache serão armazenados. Você poderá definir outros diretórios, bastando para tal, incluir novas *tags cache_dir*.

O padrão é:
```
# ones with no max-size specification last.
#
#Default:
# cache_dir ufs /usr/local/squid/var/cache 100 16 256
```

Caso não seja necessário mexer nesta configuração, não é necessário habilitar a linha

Valor da Tag	Descrição
cache_dir	Define os valores do diretório de cache utilizado pelo Squid.

ufs	O Squid assume a utilização de um sistema de arquivos Unix (ufs) para o sistema de armazenamento do cache.
/usr/local/squid/var/cache	O diretório de todos os objetos armazenados no cache será neste diretório.
100	Quantidade de dados armazenados que o Squid colocará no diretório de cache. O tamanho padrão do cache é de 100 MB.
16	Define o número de subdiretórios a serem criados no cache. O Squid divide os objetos armazenados no cache entre esses subdiretórios para agilizar o acesso aos dados. Por exemplo, o Squid encontrará um objeto muito mais rapidamente buscando em diversos diretórios no cache, em vez de buscar em um único diretório com centenas de milhares de objetos.
256	Define o número de subdiretórios secundários a serem criados no cache. Se o seu cache for extremamente grande, você deverá aumentar estes valores. Para a maioria das implementações, estes valores de subdiretório são suficientes.

TAG ACL

Esta *tag* permite a definição de uma lista de acesso. A lista de acesso pode conter endereços de IP de clientes, uma faixa de endereços, o endereço de um servidor de URL, endereço de uma máquina local ou domínios. Qualquer lista de acesso que você definir utilizando esta *tag* poderá ser utilizada mais tarde para permitir ou negar pedidos ao servidor de cache.

Vamos configurar o Squid para negar sites com a palavra “sexo” na URL:

Onde está:

```
#acl fileupload req_mime_type -i ^multipart/form-data$
#acl javascript rep_mime_type -i ^application/x-javascript$
#
#Recommended minimum configuration:
acl all src 0.0.0.0/0.0.0.0
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 563 # https, snews
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT
```

Deixe:

```
#acl fileupload req_mime_type -i ^multipart/form-data$
#acl javascript rep_mime_type -i ^application/x-javascript$
#
#Recommended minimum configuration:
acl all src 0.0.0.0/0.0.0.0
acl blockedsites url_regex -i “/usr/local/squid/etc/sitesblock.txt”
acl unblockedsites url_regex -i “/usr/local/squid/etc/sitesunblock.txt”
acl manager proto cache_object
acl localhost src 127.0.0.1/255.255.255.255
acl to_localhost dst 127.0.0.0/8
acl SSL_ports port 443 563
acl Safe_ports port 80 # http
acl Safe_ports port 21 # ftp
acl Safe_ports port 443 563 # https, snews
```

```
acl Safe_ports port 70 # gopher
acl Safe_ports port 210 # wais
acl Safe_ports port 1025-65535 # unregistered ports
acl Safe_ports port 280 # http-mgmt
acl Safe_ports port 488 # gss-http
acl Safe_ports port 591 # filemaker
acl Safe_ports port 777 # multiling http
acl CONNECT method CONNECT
```

Dentro do arquivo *sitesblock.txt* você colocará as palavras e sites que deseja bloquear, no arquivo *sitesunblock.txt* você coloca os sites que deseja autorizar. Por exemplo, se você colocar a palavra “sexo” dentro do arquivo *sitesblock.txt*, estará bloqueandos sites como www.sexo.com.br e www.sexoesaude.com.br. No caso de você querer liberar algum site (como www.sexoesaude.com.br), você deve colocar dentro do arquivo *sitesunblock.txt* o site a ser liberado.

Você deve configurar outras *tags* para garantir e melhorar o controle de sites bloqueados (para evitar que alguém acesse a Internet sem passar pelo *proxy*). Abaixo será listado o que deve ser feito, qualquer dúvida consulte o manual do software para maiores referências a respeito das *tags*.

Tag httpd_accel_host e httpd_accel_port

Onde está:

```
# HTTPD-ACCELERATOR OPTIONS
# -----------------------------------------------------------------------------
#  TAG: httpd_accel_host
#  TAG: httpd_accel_port
#    If you want to run Squid as an httpd accelerator, define the
#    host name and port number where the real HTTP server is.
#
#    If you want IP based virtual host support then specify the
#    hostname as "virtual". This will make Squid use the IP address
#    where it accepted the request as hostname in the URL.
#
#    If you want virtual port support then specify the port as "0".
#
#    NOTE: enabling httpd_accel_host disables proxy-caching and
#    ICP.  If you want these features enabled also, then set
#    the 'httpd_accel_with_proxy' option.
#
#Default:
# httpd_accel_port 80
```

Deixe:

```
# HTTPD-ACCELERATOR OPTIONS
# -----------------------------------------------------------------------------
#  TAG: httpd_accel_host
#  TAG: httpd_accel_port
#    If you want to run Squid as an httpd accelerator, define the
#    host name and port number where the real HTTP server is.
#
#    If you want IP based virtual host support then specify the
#    hostname as "virtual". This will make Squid use the IP address
#    where it accepted the request as hostname in the URL.
#
#    If you want virtual port support then specify the port as "0".
#
#    NOTE: enabling httpd_accel_host disables proxy-caching and
#    ICP.  If you want these features enabled also, then set
#    the 'httpd_accel_with_proxy' option.
#
#Default:
httpd_accel_host virtual
httpd_accel_port 80
```

Tag httpd_accel_with_proxy

Onde está:

```
#  TAG: httpd_accel_with_proxy   on|off
#    If you want to use Squid as both a local httpd accelerator
#    and as a proxy, change this to 'on'. Note however that your
#    proxy users may have trouble to reach the accelerated domains
#    unless their browsers are configured not to use this proxy for
#    those domains (for example via the no_proxy browser configuration
#    setting)
#
#Default:
#httpd_accel_with_proxy off
```

Deixe:

```
#  TAG: httpd_accel_with_proxy   on|off
#    If you want to use Squid as both a local httpd accelerator
#    and as a proxy, change this to 'on'. Note however that your
#    proxy users may have trouble to reach the accelerated domains
#    unless their browsers are configured not to use this proxy for
#    those domains (for example via the no_proxy browser configuration
#    setting)
#
#Default:
httpd_accel_with_proxy on
```

Tag httpd_accel_uses_host_header

Onde está:

```
#  TAG: httpd_accel_uses_host_header  on|off
#    HTTP/1.1 requests include a Host: header which is basically the
#    hostname from the URL.  The Host: header is used for domain based
#    virutal hosts. If your accelerator needs to provide domain based
#    virtual hosts on the same IP address then you will need to turn this
#    on.
#
#    Note that Squid does NOT check the value of the Host header matches
#    any of your accelerated server, so it may open a big security hole
#    unless you take care to set up access controls proper.  We recommend
#    that this option remain disabled unless you are sure of what you
#    are doing.
#
#    However, you will need to enable this option if you run Squid
#    as a transparent proxy.  Otherwise, virtual servers which
#    require the Host: header will not be properly cached.
#
#Default:
# httpd_accel_uses_host_header off
```

Deixe:

```
#  TAG: httpd_accel_uses_host_header  on|off
#    HTTP/1.1 requests include a Host: header which is basically the
#    hostname from the URL.  The Host: header is used for domain based
#    virutal hosts. If your accelerator needs to provide domain based
#    virtual hosts on the same IP address then you will need to turn this
#    on.
#
#    Note that Squid does NOT check the value of the Host header matches
#    any of your accelerated server, so it may open a big security hole
#    unless you take care to set up access controls proper.  We recommend
#    that this option remain disabled unless you are sure of what you
#    are doing.
#
#    However, you will need to enable this option if you run Squid
#    as a transparent proxy.  Otherwise, virtual servers which
```

```
#    require the Host: header will not be properly cached.
#
#Default:
httpd_accel_uses_host_header on
```

TAG HTTP_ACESS

Esta *tag* permite ou nega o acesso ao Squid. Você pode permitir ou negar todos os pedidos. Também é possível permitir ou negar pedidos baseados em uma lista de acesso pré-definida. Se for removido todas as entradas *http_acess*, todos os pedidos serão permitidos por padrão.

Os clientes *proxy* não serão capazes de usar o servidor *Squid proxy-caching* até que você modifique as *tags http_acess*. Observe que recomenda-se algum nível de controle de acesso, por isso não remova todas as *tags http_acess*.

Onde está:

```
# http_access deny all
#
#Recommended minimum configuration:
#
# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager
# Deny requests to unknown ports
http_access deny !Safe_ports
# Deny CONNECT to other than SSL ports
http_access deny CONNECT !SSL_ports
```

deixe:

```
# http_access deny all
#
#Recommended minimum configuration:
#
# Only allow cachemgr access from localhost
http_access allow manager localhost
http_access deny manager
# Deny requests to unknown ports
http_access deny !Safe_ports
# Deny CONNECT to other than SSL ports
http_access deny CONNECT !SSL_ports
http_access deny blockedsites !unblockedsites
http_access allow all
```

A linha *http_access allow all* irá liberar o acesso para todos as pessoas na rede utilizarem o *proxy*.

TAG CACHE_EFFECTIVE_USER

Esta tag irá configurar o usuário que será utilizado para executar o squid no momento da inicialização do processo.

Onde está:

```
#
# If Squid is not started as root, the default is to keep the
# current UID/GID, and only the GID can be changed to any of
# the groups the user starting Squid is member of. Note that if
# Squid is not started as root then you cannot set http_port to
# a value lower than 1024.
#
#Default:
# cache_effective_user nobody
```

Deixe:

```
#
# If Squid is not started as root, the default is to keep the
```

```
# current UID/GID, and only the GID can be changed to any of
# the groups the user starting Squid is member of. Note that if
# Squid is not started as root then you cannot set http_port to
# a value lower than 1024.
#
#Default:
cache_effective_user squid
```

Carregando e Testando o Squid

Para saber se está funcionando, você deverá iniciar o serviço Squid e depois usar o programa cliente do Squid no servidor local para verificar se os dados da página Web estão sendo gravados no cache. O cliente do Squid apresenta dados à medida que são gravados no cache, e é extremamente útil no teste do funcionamento do *cache proxy* e na solução de problemas.

Inicie o proxy cache Squid com o seguinte comando:

```
# /usr/local/squid/sbin/squid -z
# /usr/local/squid/sbin/squid
# /usr/local/squid/bin/squidclient http://www.laureano.tk
```

Lembre-se que os diretórios devem pertencer ao usuário configurado anteriormente.

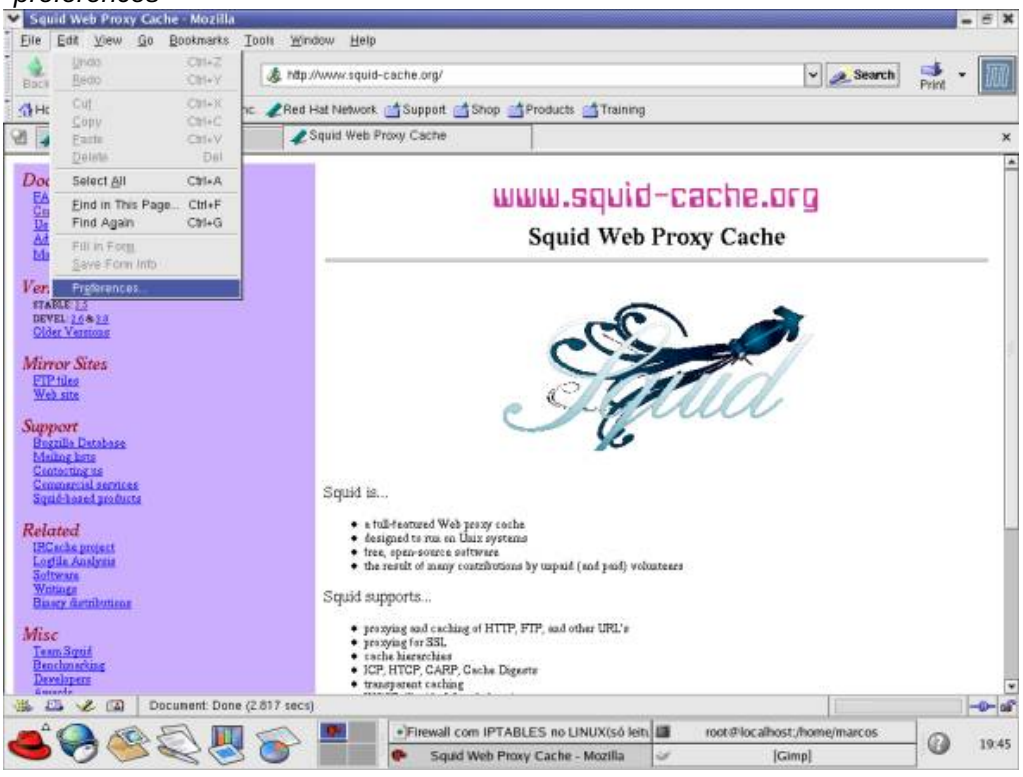
Se aparecer alguma mensagem de erro, verifique os arquivos de log que se encontram em */usr/local/squid /var/logs*.

Configuração de Clientes Proxy

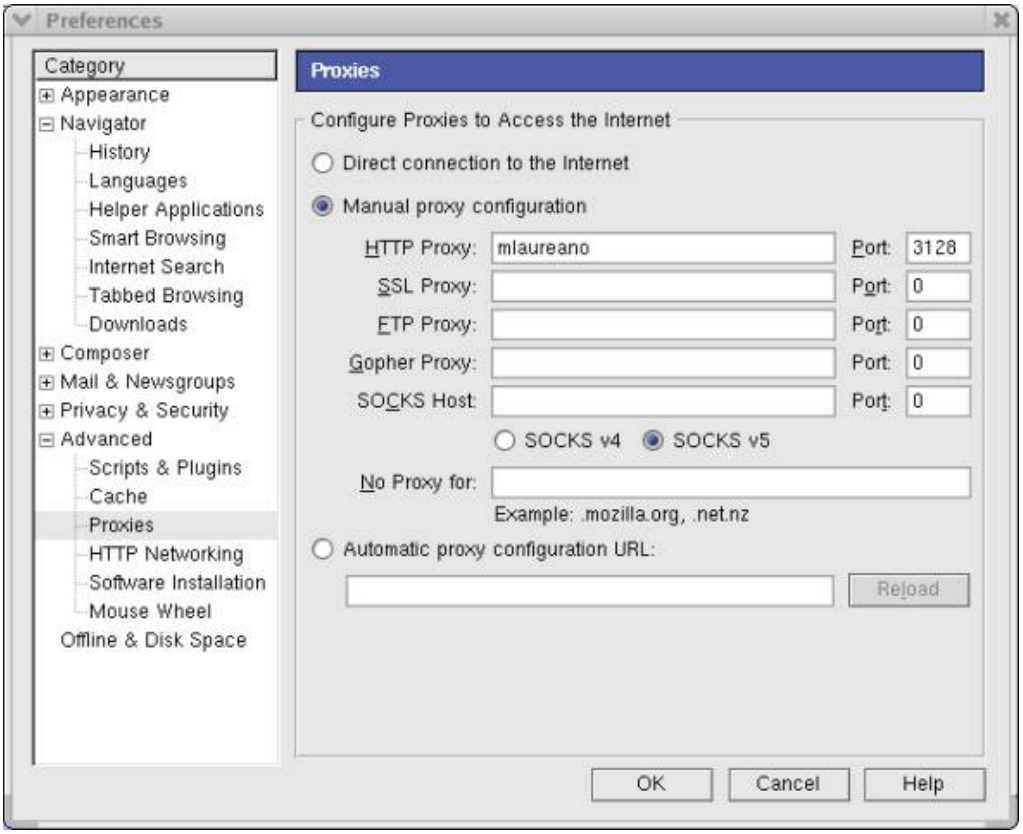
Um cliente *proxy* é um sistema que utiliza os serviços de um servidor Web de *proxy-caching*. Dependendo da configuração de sua rede, um cliente pode ou não ter que ser configurado como cliente *proxy* para poder usar um servidor *proxy cache* na Web. Por exemplo, alguns firewalls são configurados para encaminhar todo o tráfego da porta 80 que estiver saindo da rede para o servidor de proxy cache na Web. Neste caso, os clientes proxy não precisam de configuração manual. Em outros casos, o cliente detecta automaticamente a informação do servidor *proxy* na rede e a utiliza para todo o acesso à Internet.

Configurar um cliente *proxy* é muito mais fácil do que configurar o Squid. Toda a configuração do cliente *proxy* é concluída dentro do aplicativo do navegador.

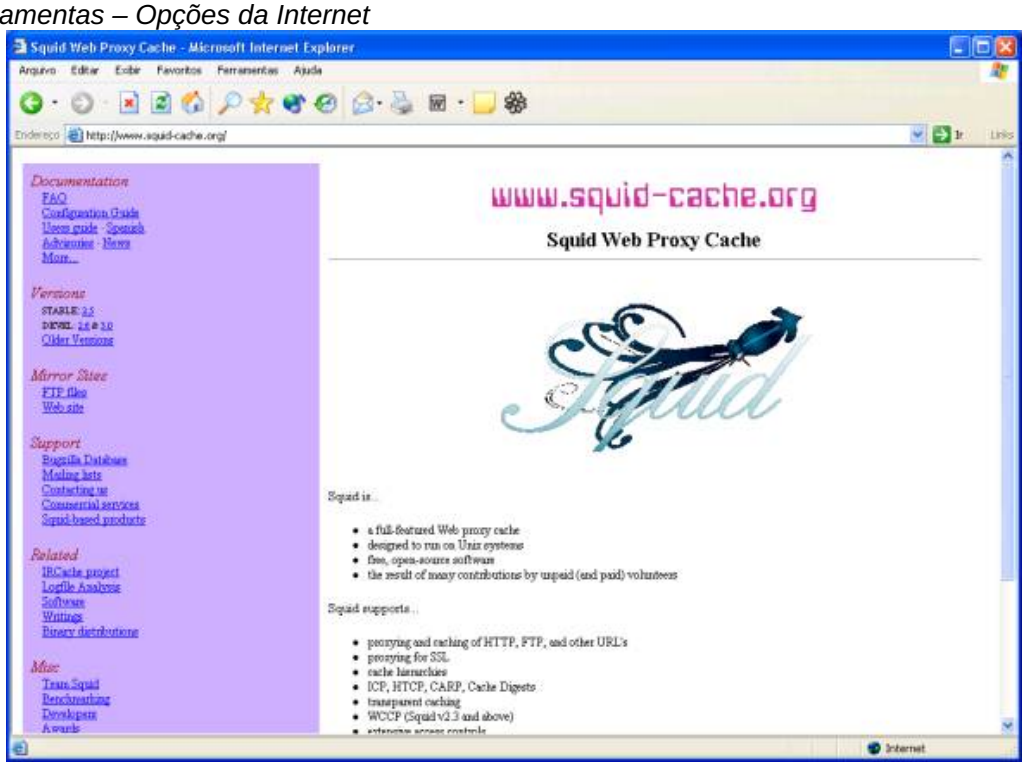
No Mozilla:
Entre em *edit - preferences*



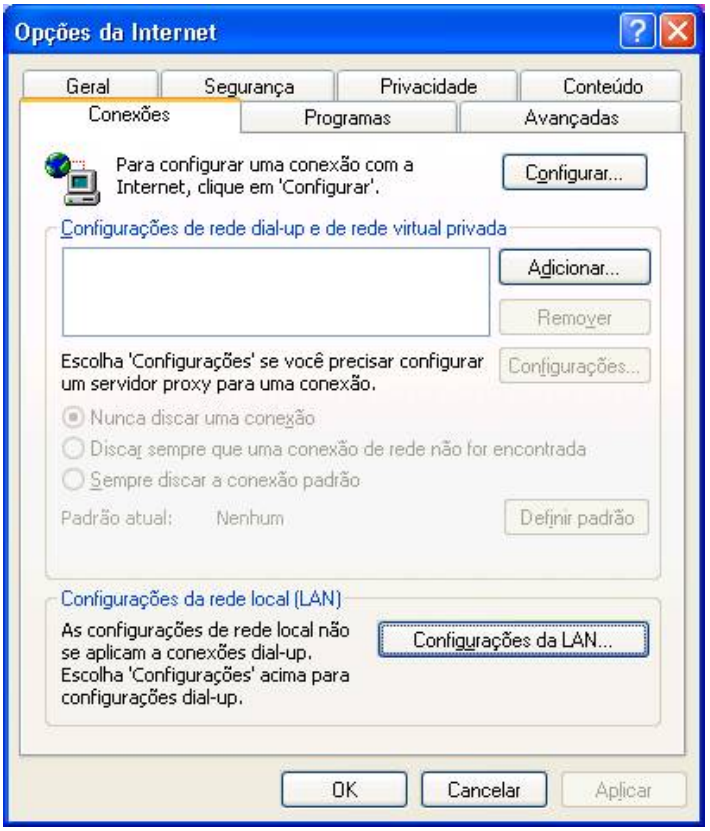
Configure no item *Advanced – Proxies.*



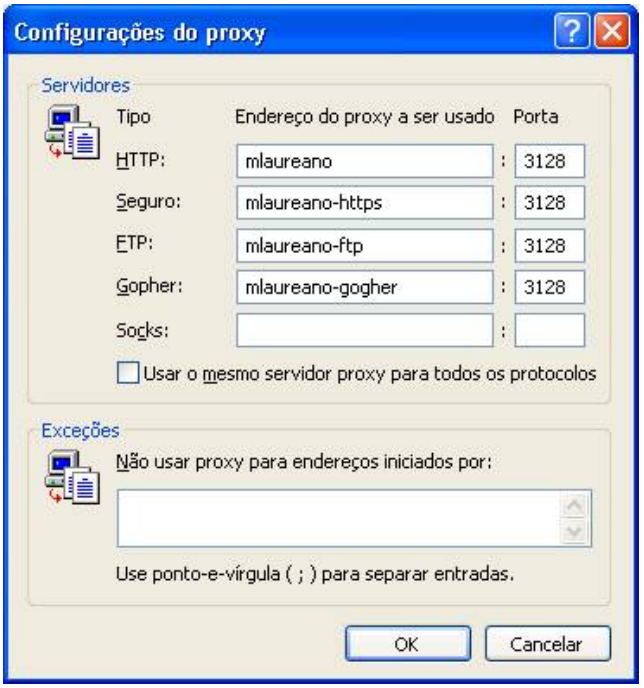
No Windows Explorer:
Entre em *Ferramentas – Opções da Internet*



Selecione a aba *Conexões* e depois clique na opção “Configurações da LAN”.



Você também pode especificar um servidor proxy para cada opção de protocolo. Clique em “Avançado”


Configurações do proxy
Servidores
Tipo
Endereço do proxy a ser usado
Porta
HTTP:
mlaureano
3128
Seguro:
mlaureano-https
3128
FTP:
mlaureano-ftp
3128
Gopher:
mlaureano-gogher
3128
Socks:
Usar o mesmo servidor proxy para todos os protocolos
Exceções
Não usar proxy para endereços iniciados por:
Use ponto-e-vírgula (;) para separar entradas.
OK
Cancelar